RESEARCH PAPER

HPC-oriented Canonical Workflows for Machine Learning Applications in Climate and Weather Prediction

Amirpasha Mozaffari[1], Michael Langguth[1], Bing Gong[1], Jessica Ahring[1], Adrian Rojas Campos[2], Pascal Nieters[2], Otoniel José Campos Escobar[3], Martin Wittenbrink[4], Peter Baumann[3] & Martin G. Schultz[1]

[1]Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

[2]Osnabrück University, 49074 Osnabrück, Germany

[3]Jacobs University Bremen, 28759 Bremen, Germany

[4]Deutscher Wetterdienst, 63067 Offenbach am Main, Germany

Keywords: FAIR; Reproducibility; Machine learning; Earth system sciences; Workflow

Citation: Mozaffari, A., et al.: HPC-oriented canonical workflows for machine learning applications in climate and weather prediction. Data Intelligence 4(2), 271-285 (2022). doi: 10.1162/dint_a_00131
Received: September 17, 2021; Revised: December 17, 2021; Accepted: February 5, 2022

ABSTRACT

Machine learning (ML) applications in weather and climate are gaining momentum as big data and the immense increase in High-performance computing (HPC) power are paving the way. Ensuring FAIR data and reproducible ML practices are significant challenges for Earth system researchers. Even though the FAIR principle is well known to many scientists, research communities are slow to adopt them. Canonical Workflow Framework for Research (CWFR) provides a platform to ensure the FAIRness and reproducibility of these practices without overwhelming researchers. This conceptual paper envisions a holistic CWFR approach towards ML applications in weather and climate, focusing on HPC and big data. Specifically, we discuss Fair Digital Object (FDO) and Research Object (RO) in the DeepRain project to achieve granular reproducibility. DeepRain is a project that aims to improve precipitation forecast in Germany by using ML. Our concept envisages the raster datacube to provide data harmonization and fast and scalable data access. We suggest the Juypter notebook as a single reproducible experiment. In addition, we envision JuypterHub as a scalable and distributed central platform that connects all these elements and the HPC resources to the researchers via an easy-to-use graphical interface.

† Corresponding author: Amirpasha Mozaffari (a.mozaffari@fz-juelich.de; ORCID: 0000-0001-6719-0425).

1. INTRODUCTION

The intention of the FAIR (Findable, Accessible, Interoperable, Reusable) principle by Wilkinson et al. [1] was not limited to data, but also targeted other Digital Objects (DO) [2], e.g., algorithms, tools and workflows that lead to data. The FAIR Digital Object (FDO) subsequently introduced by de Smedt et al. [3] provides a framework to have transparent, reusable, and reproducible data [4]. The apparent benefit of reproducible science is that it becomes possible to restore results in a critical situation, increase transparency, trust, interest and the number of citations. It can rise to a level where reusing previous work becomes a routine practice and leads to an increase in productivity, work habit, and continuity [5, 6, 7, 8]. However, the reality of science deviates from these conveyed principles. The concept of Canonical Workflow Framework for Research (CWFR) is proposed by Hardisty and Wittenburg [9] as a solution to expand the adaptation of the FAIR principle to the broader research community. CWFR relies on identifying recurring patterns across disciplines and breaking down workflows into smaller modular components that can be reused and reassembled for other use cases. In this paper, we suggest to use Jupyter notebooks on JuypterHub with connection to an HPC system [10] as a platform to develop a concept for bringing the components of the CWFR together for the application of Machine Learning (ML) in Earth System Sciences (ESS). To develop a functioning CWFR, identifying the challenges and practices of that particular community is essential. ESS in general and climate and weather, in particular have seen significant growth in recent years, thanks to petabyte-size data and exponential increase in computational capability [11, 12]. With a growing amount of data and in light of climate change including its impacts, FAIRness in ESS ensures comprehensible and reliable knowledge of the environment. However, in the particular domain of Earth sciences, more than 60% of surveyed researchers stated that they failed to reproduce someone else's experiment, while more than 40% admitted that they were unable to reproduce their own experiment [13]. The issues above also exist in ML as documented in ML conference publications. At the prestigious Conference on Neural Information Processing Systems (NIPS) in 2017, less than 40% of the publications provided links to the code. As a consequence, some studies highlight the importance of reproducible ML that allows others to apply the contributions and increase the impact of ML research [14]. The data-driven nature of the ML poses unique challenges regarding reproducibility. As more and more data is being used as *training* and *test data*, ensuring that presented results are sound and reliable is a significant challenge [8]. In addition, the training process involves *randomness*. For instance, stochastic gradient descent (which is widely used for ML model updates) uses a randomised procedure that could result in different weights at each run even though an identical code is used [15]. Furthermore, ML frameworks are commonly used to speed up development. The mainstream frameworks such as TensorFlow [16] and PyTorch [17] use mixed precision for accelerating GPUs' training process that could yield different results depending on underlying software or hardware. Furthermore, most ML algorithms use a vast range of libraries and frameworks, configurations and virtual environments that rapidly change so that other versions can lead to different outputs. Challenges mentioned in ESS and ML are compounding when ML methods are applied to ESS data. ESS and ML algorithms rely on large data volumes which require rapid processing and thus high-performance computing (HPC) resources. Well-designed workflows play an important role in handling elaborate data preparation and efficiently utilising tomorrow's exascale computers. The widespread adaptation of workflow applications face two significant challenges: a vast gap between workflow applications utilised in enterprise-scaled IT

firms and science labs and shared beliefs that researcher's applications are unique [9]. More than 90% of the researchers surveyed by Stoddart [13] agreed with "more robust experimental design" as a necessity for enhanced reproducibility of scientific results. A well-designed workflow that ensures reproducibility and traceability in every step could greatly enhance the robustness of ESS experiment design. Having reusable software and workflow components doesn't imply that individual scientific ideas can no longer be pursued. By contrast, they will form a solid reproducible basis on which new ideas can build with less potential for errors.

Inspired by the expected benefits of integrating reproducibility in ML applied to ESS, we discuss the particular challenges in our interdisciplinary project, DeepRain, in Section 2. This project is an excellent example of the requirement of interoperable and reproducible research as it aims to improve precipitation forecast using ML. In this context, the benefits of FAIRification on ML applications in ESS are discussed, and we describe how existing concepts and methods can be used in Section 3. Section 4 then introduces our proposed framework based on CWFR, which aims to provide flexible and reproducible ML focusing on big data analysis on HPC systems. A conclusion and an outlook based on our concept are given in the final Section 5.

2. PROBLEM FORMULATION AND PREREQUISITES

The DeepRain project serves as a particular research example for which a canonical workflow framework is crucial. In the DeepRain project [18], more than 1.3 PBytes of meteorological data are exploited to develop complex ML algorithms to predict precipitation in Germany. In particular, historical forecasts of the COSMO-DE (Consortium for Small-Scale Modelling) Ensemble Prediction System (EPS) provided by the German Weather Service (DWD) [19, 20, 21] serve as the input for quantitative precipitation forecasts at station sites and on gridded domains based on tailor-made deep neuronal networks. For the latter, the high-resolution radar-based climatology product RADKLIM acts as a high-quality observational reference dataset [22, 23]. Processing a large amount of data requires massive computational resources that are exclusively available on the HPC systems. Because of bandwidth limitations, the bigdata is usually hosted in the same facility that provides the HPC system to reduce data streaming delay and connection interruptions. So, any suggested solution should provide easy access to HPC systems, stored data, and in-situ processing to avoid unnecessary uploads and downloads of data. In the DeepRain project, where a relatively broad group of researchers from ML, weather and climate and computer sciences (CS) are collaborating, we need a framework that can allow for efficient collaboration and does not require too much CS expertise. The project requires using technical tools, communicating and developing ML models and experiments while reducing the time and energy necessary to get acquainted with methods and terminology used in different disciplines. A FAIR practice helps to make the research interoperable across other disciplines, even in the same project and group. One main obstacle to adopt FAIR practices is that many of them require drastic changes in the researcher's procedures. Thus, any suggested solution should adjust to the researcher's needs rather than being constrained by IT considerations [24]. The changes in the researcher's work should be minimized to serve both the fulfilment of FAIR practices and the acceptance by the researcher. In addition,

any pre-designed workflow must allow the use of domain specific language and procedures, for example with respect to the evaluation of results. Therefore, we narrowed down our focus to ML applications for the ESS community; to ensure easy adaptation without extensive development. If the proposed approach is picked up by the ESS community, it can be further developed and generalised to meet the requirements of other science communities. A high degree of flexibility is necessary for any proposed CWFR to adapt to researcher practices. However, designing a flexible solution needs a certain degree of computer knowledge by the researcher. Thus, it is necessary for researchers to be familiar with CS fundamentals such as Unix, bash and version control (e.g., with git). However, much of the higher-level expertise to develop and provide services to maintain a CWFR is out of reach of arbitrary research institutions. Thus, a collaboration between service providers, such as data and HPC centres, and research communities is necessary to maintain a sustainable ecosystem. Such partnership provides expertise, training and infrastructure for research communities. Therefore, a convergence between the research communities and infrastructure providers is necessary. Even though computational experiments should be easier to reproduce compared to physical experiments, the complexities and fast pace of change of today's software and hardware make it surprisingly difficult [25]. Research needs to be reproducible for a human, referred to as *scientific reproducibility* , and for a machine, referred to as *technical reproducibility*. As mentioned above, randomised processes, mixed-precision, and hardware-dependency impediment on technological reproducibility. Thus, we focus on scientific reproducibility, where we can reproduce statistical features and underlying distributions. The latter means that the final results may deviate slightly from past experiments even with an identical set-up, but the obtained statistical properties (e.g., model performance in terms of evaluation scores) and the corresponding conclusions must remain the same. In this sense, the deviations of the obtained results must be indistinguishable from random noise.

3. FAIRNESS BUILIDING BLOCKS

In the following, we introduce the components that can help to build a FAIR ecosystem addressing the obstacles mentioned in Section 2, while reducing the cost of the FAIRification.

As Kahn and Wilensky [2] introduced the concept of DO, it provides a framework to identify and trace any digital objects such as data, algorithm, workflow. DOs, alongside Persistent Identifiers (PID) and metadata, are elements that constitute an FDO [24]. FDO as a self-contained, typed, machine-actionable data package can provide basic components for a standardised, FAIR infrastructure [3]. FDO lays out a fundament which can be used to bring FAIRness to all components of science. As the adaptation of FDO is highly domain-oriented, we refer to Lannom et al. [26] for more details on the application for ESS.

Research Object (RO) is a related concept that was introduced by Bechhofer et al. [27] where the main focus is on born-digital objects and aggregation of data and collections. Thus, RO is well suited for application in data-driven sciences. In addition, RO can be associated with DOI, thereby making it findable and accessible over the internet. RO concept relies on the idea that each RO provides a unit of knowledge.

Therefore RO acts as a container of resources (including a series of FDOs) and is shareable within and across different research groups. Our approach uses the FDO and RO as its building blocks to create a FAIR framework.

3.1 Notebook and Git

The emerging pattern of data-oriented research is to use in-situ analysis and visualisation on the HPC system. This enables researchers to act quickly based on outputs and apply modification and restart their workflows [10]. Jupyter notebook is a tool that researchers increasingly rely on [28]. Jupyter is also recognised by Hardisty and Wittenburg [9] as one possible CWFR solutions. Beg et al. [29] also discuss the reproducibility of Jupyter notebooks as a scientific workflow. Jupyter notebook provides a one-study, one-document concept that is easily shareable. In addition, it provides a user-friendly platform to document the software while ensuring reproducibility by combining the data, code and software environment. As Jupyter notebook offers the core, JupyterHub expands the frameworks and brings flexibility to the user group [30]. JupyterHub provides access control and authentication, scalability with support for container and HPC technology, and it is portable from the cloud to a local machine. Despite the benefits, there is some limitations to deploy notebooks as CWFR solution. Any modification to the notebook is immediate and multiple executions of a notebook with different inputs leads to loss of all previous information. In addition, any part of the notebook can be executed or skipped separately. It is known as a problem of *undefined state* of the notebook. Thus, the constant prototyping and rapid development ecosystem of Jupyter notebook threatens its adaptation as reproducible workflow [29]. Changes applied to notebooks usually fall into two categories of 1) code developments to introduce new features and methods or 2) experimenting in the hyperparameter space. The primary tool to keep track of changes in algorithms will remain version control, particularly git [31]. As there are many comprehensive studies about the application of git for reproducibility, we refrain from repeating and refer the reader to Ram [32]. Each newly committed snippet of code is identified with a unique commit-ID. Thus, we built our approach around git. ML applications often need to experiment with the parameter space as it requires intensive hyperparameter optimisation and search of model space. It is essential to preserve the notebook state and each experiment parameter. We propose an application of a notebook that we call *experiment dashboard* that is used to initiate any experiment notebook. Dashboard configuration, including a summary of all carried out experiments and paths to data and associated commit-ID for the current instance of the dashboard, is stored as an FDOs. For executing individual instances of the notebook separately, we suggest a python library called papermill that can run the new instance with parameter values passed to it. Any new experiment with new parameter values is passed to a new instance of the notebook, and all will be preserved. As shown in Figure 1, the experiment dashboard passes the desired three values to three independent instances of the notebook. Each of the instances is executed individually and the instances can also be run in parallel.

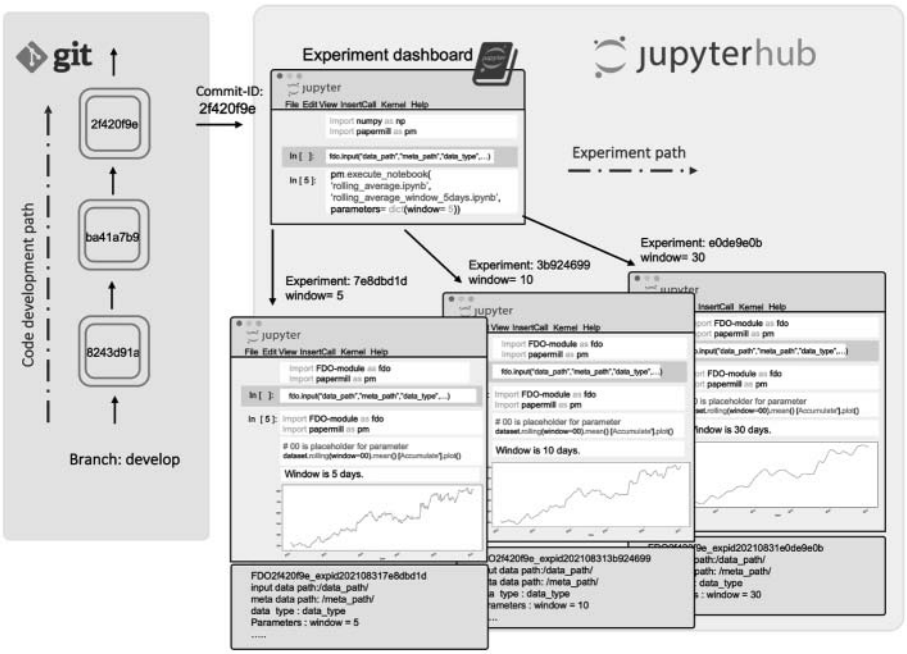

Figure 1. Integration of git and experiment dashboard utilising papermill to execute each experiment in an individual instance of the notebook with passed parameters and the creation of corresponding FDO with FDOm.

3.2 FDO and RO Modules

As there are no standard or widely used FDO libraries so far, a dedicated FDO module (FDOm) is envisioned to ensure the proper creation of FDO. In the following, we define principle rules for FDOs created by FDOm. Every unique experiment generates one FDO which points to one commit-ID generated by git. FDOs can either point to data or to another FDO; this referencing is called *interlocking FDO*. Any interlocking FDO appends all the locked FDOs in the one new FDO. Since FDOs are only backwards-looking, they can only be linked to data, commit or other FDOs that exist by the time of creation. Besides, they include metadata and are searchable. In addition, we introduced technical regulations as well. We expect that the FDO provides information about the host system that is used. This contains system configuration data that is provided by the system admin and the environment as well as libraries detected by FDOm. For FDOm to be useful for ML applications, it is essential to document specific ML architecture and initial parameters. As TensorFlow [16] and PyTorch [17] are the main ML frameworks used in our project, FDOm will be tailored to receive the network summary directly. When an experiment ends, a unique PID is generated to identify the created FDO. FDOm is storing this information as JSON-LD which is human and machine-readable. As shown in Figure 1, each instance of the notebook experiment is associated with a unique FDO. Furthermore, we envision a RO module, called ROm subsequently, to create the necessary RO which may encapsulate several FDOs. Similar to FDOm we foresee principle rules for ROm. Each RO has a state attribute which can be *open, archived* or *published*. An open RO is mutable

and a new FDO can be added to it. An archived RO has been packaged, and is is therefore immutable. A published RO is similar to an archived one, except that a DOI is assigned to it. Any new RO can be created from scratch or based on an archived or published RO. The ROm allows the researcher to inspect other involved FDOs and to select individual ones for later use. For example, any individual ML experiment can be selected and encapsulated as a new RO. We suggest preserving all experiments and their output regardless of whether they have been successful or not or whether they are intended for publication or not. In contrast to FDO, the RO concept is already quite well developed and many implementations exist. From many implementations of the RO, we adapt our proposed ROm to be compatible with Ro-Crate [33] . We believe that the combination of FDOm and ROm as well as their integration with git and RO-Crate could provide the necessary ecosystem to achieve high level granular reproducibility.

3.3 Datacube Management

To address the efficient data management challenge in the DeepRain project, we deployed an array-centric database. After evaluating several candidates, we opted for the Array Database Management System rasdaman [34] which offers geo-semantic query functionalities for multidimensional arrays, also referred to as *datacubes* [35]. In parallel to traditional file-based data, rasdaman provides efficient management of large-scale objects, and standardized data modeling [36] which contributes to data harmonization and it allows for flexible access, extraction, analysis, and fusion of massive Spatio-temporal datacubes based on a standardized query language. Figure 2 shows an exemplary query used to retrieve data from Deep Rain datacube. Data could be requested as a query while the access pattern is registered in the related FDO.

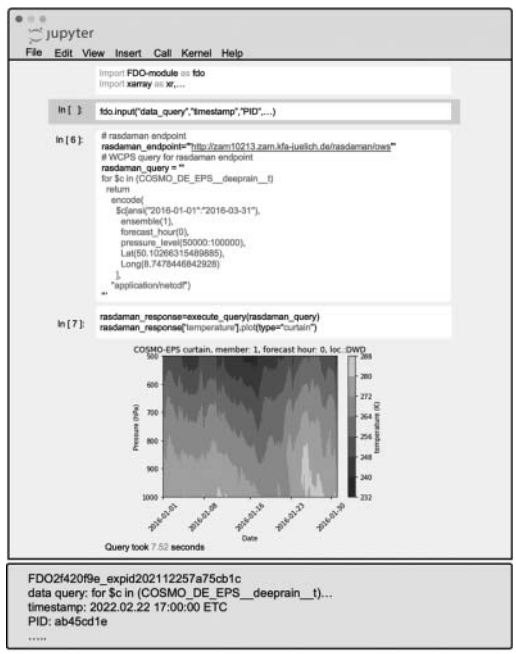

Figure 2. Example query from DeepRain datacube.

4. THE PROPOSED FRAMEWORK

In this section, we propose a framework built on the tools mentioned previously or modified to fit the research needs in ML application in ESS. Our proposed concept relies on a granular approach that uses FDO and RO as its building blocks. Every single ML experiment is registered as an FDO. The method that is used to produce the training data should be available to achieve reproducibility, complemented by explicit descriptions of the used pipeline and architecture. A series of experiments with corresponding FDOs is then encapsulated in a RO. This method ensures that the entirety of the research remains reproducible and that FAIRness is not limited to publication. Besides, to achieve a granular FAIRness, we suggest a FAIR-test similar to a unit-test as standard practice in CS where small segments of codes such as function, method, class, etc., are tested. The same principle can be integrated into research practices to validate the FAIRness of each segment of carried out research. Our suggested CWFR is built around the JupyterHub as a platform that glues notebooks, FDO and RO together. Jülich Supercomputing Centre (JSC) implemented an instance of JupyterHub called Jupyter-JSC[①] that provides a suitable platform for our proposed framework [10]. Jupyter-JSC has access to a wide range of data storage, to CPU and GPU on the JUWELS [37] and other HPC systems, and provides an easily accessible integration with git. The proposed framework is presented in the following prototype scenario. As the researcher develops codes, git provides a perfect avenue for preserving them and reusing them in the Jupyter notebook instance running on the HPC infrastructure. As one wants to run ML experiments, the experiments dashboard is used to initiate them. Required data is accessed via a path to the data on the local file system of the HPC system, to an online repository, or via a query to the datacube interface. The experiment dashboard then creates several Jupyter notebook instances according to the number of experiments while also passing the ML experiments parameters. Each model will call the FDOm that sets up the associated FDO for all unique experiments, including data (path), summary ML architectures, random seeding generated, etc. The relevant local copies of downloaded results are also stored and tracked. Any subsets of FDOs associated with ML experiments can be encapsulated as RO with the help of the ROm. The created RO can be used by the researchers to share a holistic view of their work with collaborators, to archive them, to reuse or to continue working based on previous achievements. The work has begun to implement the proposed concept, and we plan to derive a more concrete scheme as a follow-up technical paper.

① https://jupyter-jsc.fz-juelich.de/

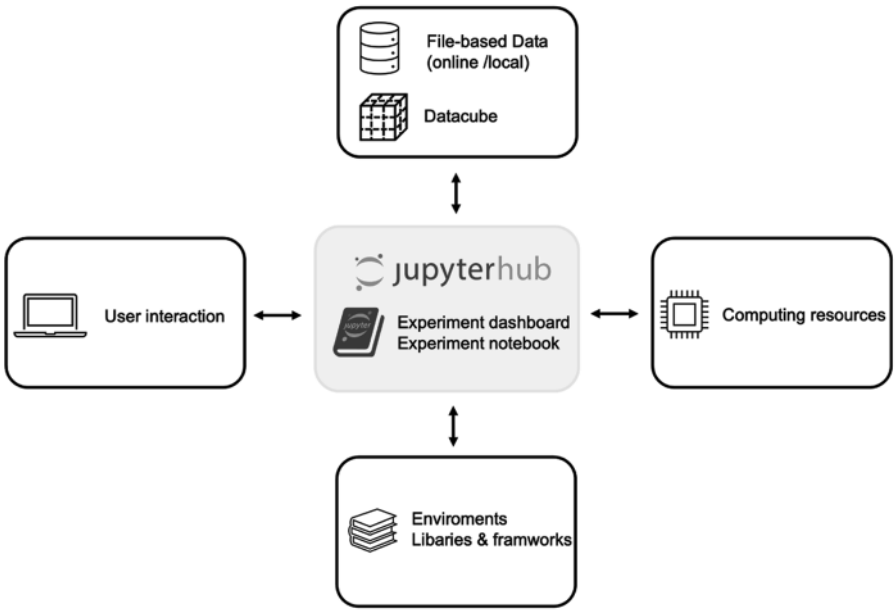

Figure 3. JupyterHub is a platform that provides access, authentication and interface to the user, data, environments and frameworks and the computing resources in the HPC system.

5. CONCLUSION AND FUTURE WORK

We discussed necessary elements of a canonical workflow for ML applications in ESS to ensure FAIR and reproducible research while exploiting the HPC capability. Our solution builds upon on FDO and RO, and we envision a concept of FAIR unit-test where a researcher can validate the FAIR practices for small segments of codes and experiments. We have introduced some basic rules that ensure that FDO and RO are human- and machine-actionable and that they can achieve scientific reproducibility. For this, we proposed two modules of FDOm and ROm to enforce the suggested basic rules without introducing unnecessary changes to the researcher workflow. The modeules ensure that each experiment is identified by a unique FDO and that series of experiments are encapsulated as RO. In addition to file-based data storage, datacubes provide quick access to data with an integrated FDO pointer function. We have proposed the Jupyter notebook as the core of the CWFR while acknowledging its limitation in a particular undefined state of a notebook. We suggest an experiment dashboard where the researcher can initiate new experiments as an independent notebook. Papermill is a Python-based library that allows us to preserve and document changes in each notebook independently. The approach presented in this study aims to minimize technological barriers for ESS researchers to shift toward integrated FAIR practices. Nevertheless, elevating the fundamental knowledge and skills in CS should remain a goal for ESS communities, because CS developed many concepts and tools to ensure versioning, tracking, reproducibility and portability. These tasks constitute the backbone of FAIR and reproducible research and are the pillars on which canonical workflows can be built.

AUTHOR CONTRIBUTIONS

A. Mozaffari (a.mozaffari@fz-juelich.de) has initiated the concept study and developed the core ideas in discussions with M.G. Schultz (m.schultz@fz-juelich.de), M. Langguth (m.langguth@fz-juelich.de) and B. Gong (b.gong@fz-juelich.de). All authors have made meaningful and valuable contributions to writing and revising the manuscript. A. Mozaffari has led the editorial process.

ACKNOWLEDGEMENTS

The authors would like to thank German Bundesministerium fuer Bildung und Forschung (BMBF) for funding the DeepRain project under grant agreement 01 IS18047A-E. The authors gratefully acknowledge the Earth System Modelling Project (ESM) for funding this work by providing computing time on the ESM partition of the supercomputer JUWELS [37] at the Jülich Supercomputing Centre (JSC). The authors would like to thank the two anonymous reviewers and the guest editor for suggestions and comments that significantly improved the manuscript.

REFERENCES

[1] Wilkinson, M.D., et al.: Comment: The FAIR guiding principles for scientific data management and stewardship. Scientific Data 3, Article No. 160018 (2016)
[2] Kahn, R., Wilensky, R.: A framework for distributed digital object services. International Journal on Digital Libraries 6(2), 115–123 (2006)
[3] De Smedt, K., Koureas, D., Wittenburg, P.: FAIR digital objects for science: From data pieces to actionable knowledge units. Publications 8(2), Article No. 21 (2020)
[4] Lamprecht, A.-L., et al.: Towards FAIR principles for research software. Data Science 3(1), 37–59 (2019)
[5] Sandve, G. K., et al.: Ten simple rules for reproducible computational research. PLoS Computational Biology 9(10), e1003285(2013)
[6] Gil, Y., et al.: Toward the geoscience paper of the future: Best practices for documenting and sharing research from data to software to provenance. Earth and Space Science 3(10), 388–415 (2016)
[7] Donoho, D.L.: An invitation to reproducible computational research. Biostatistics 11(3), 385–388 (2010)
[8] Pineau, J., et al.: Improving reproducibility in machine learning research (A report from the NEURIPS 2019 Reproducibility Program). arXiv preprint arXiv: 2003.12206v2 (2020)
[9] Hardisty, A., Wittenburg, P.: Canonical Workflow Framework for Research (CWFR) - position paper - version 2 December 2020. Working paper. Available at: https://osf.io/9e3vc/. Accessed 9 December 2021
[10] Gobbert, J.H., et al.: Enabling interactive supercomputing at JSC Lessons Learned. In: ISC High Performance 2018: High Performance Computing, pp. 669–677 (2018)
[11] Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature 525(7567), 47–55 (2015)
[12] Reichstein, M., et al.: Prabhat, deep learning and process understanding for data-driven earth system science. Nature 566 (7743), 195–204 (2019)
[13] Stoddart, C.: Is there a reproducibility crisis in science? Nature (2016). Available at: https://doi.org/10.1038/d41586-019-00067-3. Accessed 9 December 2021
[14] Tatman, R., Vanderplas, J.: A practical taxonomy of reproducibility for machine learning research. In: The 2nd Reproducibility in Machine Learning Workshop at ICML 2018. Available at: https://hub.docker.com/r/pangwei/tf1.1/. Accessed 9 December 2021

[15] Beam, A.L., Manrai, A.K., Ghassemi, M.: Challenges to the reproducibility of machine learning models in health care. JAMA 323(4), 305–306 (2020)
[16] Developers, T.: TensorFlow. Available at: https://doi.org/10.5281/zenodo.5095721. Accessed 9 December 2021
[17] Paszke, A.: et al.: PyTorch: An imperative style, high-performance deep learning library. In: The 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), pp. 1–12 (2019)
[18] DeepRain. Available at: https://www.deeprain-project.de/. Accessed 9 December 2021
[19] Doms, G., et al.: A description of the nonhydrostatic regional COSMO model. Part II: Physical parameterization, consortium for small scale modelling. Deutscher Wetterdienst. Available at: https://doi.org/10.5676/DWD_pub/nwv/cosmo-doc_5.00_II. Accessed 9 December 2021
[20] Gebhardt, C., et al.: Uncertainties in COSMO-DE precipitation forecasts introduced by model perturbations and variation of lateral boundaries. Atmospheric Research 100(2-3), 168–177 (2011)
[21] Peralta, C., et al.: Accounting for initial condition uncertainties in COSMO-DE-EPS. Journal of Geophysical Research: Atmospheres 117 (D7) (2012). Available at: https://doi.org/10.1029/2011JD016581. Accessed 9 December 2021
[22] Winterrath, T., et al.: Erstellung einer radargestützten Niederschlagsklimatologie. Available at: https://refubium.fu-berlin.de/handle/fub188/21892. Accessed 9 December 2021
[23] Winterrath, T., et al.: An overview of the new radar-based precipitation climatology of the Deutscher Wetterdienst—data, methods, products. In: Rainfall Monitoring, Modelling and Forecasting in Urban Environment. UrbanRain18: The 11th International Workshop on Precipitation in Urban Areas, pp. 132–137 (2019)
[24] Wittenburg, P., Strawn, G.: Common patterns in revolutionary Infrastructures and data. EUDAT (2018). Available at: http://doi.org/10.23728/b2share.4e8ac36c0dd343da81fd9e83e72805a0. Accessed 9 December 2021
[25] Ivie, P., Thain, D.: Reproducibility in scientific computing. ACM Computing Surveys 51(3), Article No. 63 (2019)
[26] Lannom, L., Koureas, D., Hardisty, A.R.: FAIR data and services in biodiversity science and geoscience. Data Intelligence 2(1-2), 122–130 (2020)
[27] Bechhofer, S., et al.: Research objects: Towards exchange and reuse of digital knowledge. Nature Precedings (2010). Available at: https://doi.org/10.1038/npre.2010.4626.1. Accessed 9 December 2021
[28] Weigel, T., et al.: Making data and workflows findable for machines. Data Intelligence 2(1-2), 40–46 (2020)
[29] Beg, M., et al.: Using Jupyter for reproducible scientific workflows. Computing in Science and Engineering 23(2), 36–46 (2021)
[30] Cholia, S., et al.: Towards interactive, reproducible analytics at scale on HPC system. In: Proceedings of UrgentHPC 2020: 2020 International Workshops on Urgent and Interactive HPC, pp. 47–54 (2020)
[31] Chacon, S., Straub, B.: Pro Git: Everything you need to know about Git, Apress. Second ed. Available at: https://git-scm.com/book/en/v2. Accessed 9 December 2021
[32] Ram, K.: Git can facilitate greater reproducibility and increased transparency in science. Source Code for Biology and Medicine 8(1), Article No. 7 (2013)
[33] Soiland-Reyes, S., et al.: Packaging research artefacts with RO-Crate. arXiv preprint arXiv: 2108.06503 (2021)
[34] Baumann, P., et al.: The multidimensional database system. RasDaMan 27, 575–577 (1998)
[35] Baumann, P., et al.: Array databases: Concepts, standards, implementations. Journal of Big Data 8, Article No. 28 (2021)
[36] Baumann, P.: The OGC Web coverage processing service (WCPS) standard. GeoInformatica 14, 447–479 (2010)
[37] Support, S.: JUWELS: Modular Tier-0/1 supercomputer at Jülich Supercomputing Centre. Journal of Large-scale Research Facilities 5, 1–8 (2019)

AUTHOR BIOGRAPHY

Amirpasha Mozaffari is a postdoctoral researcher of the group on Earth System Data Exploration (ESDE) at the Jülich Supercomputing Centre (JSC) part of Forschungszentrum Jülich. He is trained as a geoscientist and recently defended his Ph.D. in Computational Geohydrophysics from RWTH Aachen. He is active in the field of data management, workflow design and FAIR data practices. He is co-chair of the canonical workflow framework for research in the Fair Digital Object forum.
ORCID: 0000-0001-6719-0425

Michael Langguth received his Master's degree in Meteorology in January 2016 at the Rheinische Friedrich-Wilhelms University of Bonn. During his Ph.D., he integrated and developed a hybrid parameterization scheme for convection into the ICON model, the operational weather prediction model by the German Weather Service DWD. In March 2020, he joined the Earth System Data Exploration (ESDE) group at Jülich Supercomputing Centre (JSC). Since then, he has developed neural networks for meteorological applications which inherently also include conceptualization of workflows in context of deep learning.
ORCID:0000-0003-3354-5333

Bing Gong received her Ph.D. degree from the Technical University of Madrid, Spain, in 2017. From 2017 to 2018, she worked as a Data Scientist in Corning Incorporated, Shanghai. Since January of 2019, she has been working in the Earth System Data Exploration (ESDE) group at Jülich Supercomputing Center (JSC). Her current research interest is deep learning for earth science applications, and machine learning software development.
ORCID: 0000-0001-7770-2738

Jessica Ahring is working in the Earth System Data Exploration (ESDE) group at Jülich Supercomputing Centre (JSC). She is studying MSc in Computer Science at RWTH Aachen with solid experience in database and HPC systems.
ORCID: 0000-0002-1227-379X

Adrian Rojas Campos has been a Ph.D. student in Cognitive Science at the Osnabrück University, Germany since 2019. In 2017 he obtained his BSc in Psychology and in 2019 a MSc degree in Cognitive Science from Osnabrück University. His area of interest is the integration of deep learning algorithms in the research processes of multiple sciences.
ORCID: 0000-0002-9036-1031

Pascal Nieters is a Ph.D. student in the Neuroinformatics Lab at the Institute for Cognitive Science at Osnabrück University where he works on computational models of neural computation. He develops models that help understand how the neural tissue in brains physically solves computational problems and how this understanding may be transferred to develop new computer hardware. He also applies neural network models as a machine learning tool to help understand and predict the behaviour of dynamical systems.
ORCID: 0000-0003-0538-6670

Otoniel José Campos Escobar is a Ph.D. candidate at Jacobs University, Bremen. His research focuses on the integration of linear algebra in array databases for machine learning applications. He has been working as researcher and software engineer in the DeepRain project since July, 2020 in the integration of weather data and datacubes using the array database rasdaman.
ORCID: 0000-0003-0656-3747

Martin Wittenbrink is a Scientist at the German Meteorological Service DWD. He earned his Master of Science degree in Physics of the Earth and Atmosphere at the University of Bonn in 2020. He is currently part of the research project DeepRain and develops spatial precipitation postprocessing methods.
ORCID: 0000-0002-6227-7609

Dr. **Peter Baumann** is Professor of Computer Science, inventor, and entrepreneur. He received his Ph.D. with "summa cum laude" from Technical University Darmstadt, Germany in 1993. At Jacobs University, Bremen, Germany he researches on flexible services for massive multi-dimensional datacubes and their application in science and engineering. In this field he has published over 160 book chapters, journal, and conference articles and has internationally patented the datacube technology. With the rasdaman Array Database system, which is proven on Petabyte-scale spatio-temporal databases and across thousands of cloud nodes, he has pioneered the field of array databases and datacubes. For its successful commercialization he has founded and led hitech spinoff, Rasdaman GmbH. Rasdaman has received a series of international innovation awards, such as recently the 2019 US TechConnect Award and the 2019 DIN Innovator Award. For many years Peter Baumann has been critically shaping and often leading datacube standards in ISO, OGC, and the European legal framework for a common SDI, INSPIRE. See www.peter-baumann.org for more information.
ORCID: 0000-0003-3860-4726

PD Dr. **Martin Schultz** works at the interface between atmospheric and computer science. He obtained his Ph.D. at Forschungszentrum Jülich in 1995 and worked at Harvard University and the Max Planck Institute for Meteorology before returning to Jülich in 2006. Since 2017 he has established the research group on Earth System Data Exploration (ESDE), which develops new machine learning methods and FAIR data workflows for Earth system science at the Jülich Supercomputing Centre (JSC).
ORCID: 0000-0003-3455-774X